UCRL-JC-121813
PREPRINT

An Efficient Algorithm for Geocentric to Geodetic Coordinate Conversion

Ralph M. Toms
Advanced Simulations Projects Manager
Conflict Simulation Laboratory
Lawrence Livermore National Laboratory

This paper was prepared for submittal to the
13th Workshop on Standards for the Interoperability of Distributed Simulations
Orlando, FL
September 18-22, 1995

September 1995

An Efficient Algorithm for Geocentric to Geodetic Coordinate Conversion

Ralph M. Toms
Advanced Simulations Projects Manager
Conflict Simulation Laboratory
Lawrence Livermore National Laboratory

KEYWORDS

Coordinate Conversion/System,
Correlation,
Datum Transformation,
Modeling and Simulation Interoperability

ABSTRACT

The problem of performing transformations from geocentric to geodetic coordinates has received an inordinate amount of attention in the literature. Numerous approximate methods have been published. Almost none of the publications address the issue of efficiency and in most cases there is a paucity of error analysis. Recently there has been a surge of interest in this problem aimed at developing more efficient methods for real time applications such as DIS. Iterative algorithms have been proposed that are not of optimal efficiency, address only one error component and require a small but uncertain number of relatively expensive iterations for convergence.

In this paper a well known rapidly convergent iterative approach is modified to eliminate intervening trigonometric function evaluations. A total error metric is defined that accounts for both angular and altitude errors. The initial guess is optimized to minimize the error for one iteration. The resulting algorithm yields transformations correct to one centimeter for altitudes out to one million kilometers. Due to the rapid convergence only one iteration is used and no stopping test is needed. This algorithm is discussed in the context of machines that have FPUs and legacy machines that utilize mathematical subroutine packages.

INTRODUCTION

The problem of transforming from geodetic coordinates to geocentric coordinates has received an inordinate amount of attention for what seems to be a relatively simple problem. The author has encountered more than forty papers on the problem in the literature and has included some of the more significant ones in the references to this paper[1-7,10-18,20-26,29]. Several different types of approximate solutions are available including tabulations, series expansions and iterative approaches. A surprisingly large number of authors believe that no closed form solution exists although it is easily derived[2,3,10,11,25]. Most authors provide very little in the way of an error analysis. In some cases only the altitude errors are addressed[6,12,29]. For applications such as radio astronomy the angular errors may dominate and cannot be ignored. A notable exception to these observations is the paper by Borkowski published in 1989 that compares the accuracy of several procedures, including closed form solutions[3].

Almost none of the papers address computational efficiency. Borkowski reports run time comparisons but makes no attempt to improve efficiency because it was not an issue for his application. The closed form solutions involve the algebraic solution of a quartic equation by the classical method due to Ferrari. For closed form solutions some care must be taken to avoid computationally costly complex arithmetic and the inevitable ill conditioning that is associated with analytic solutions of this type. The ill conditioning occurs because the direct Ferrari formulation leads to the subtraction of numbers with large magnitude that have opposite signs. This in turn requires the use of multiple precision even on machines having extended word length. Borkowski shows how to formulate the quartic to avoid both the complex arithmetic and the ill conditioning. However, this reformulation introduces some relatively expensive transcendental function evaluations. In addition, the Ferrari method itself requires several relatively time consuming square root and cube root operations. As a result the closed form solutions are not very efficient.

Recent developments in real time distributed simulation, particularly the Distributed Interactive Simulation (DIS) Program[8,19], have led to a renewed interest in the problem in an attempt to attain more efficiency[12,13,29]. These papers employ essentially the same procedure to reduce the problem to a quartic equation in a single variable. In each case the resulting quartic is solved by an iterative procedure based on Newton's method. These formulations lead to algorithms that are not of optimal efficiency for the accuracy attained. In this paper one of the classic iterative methods is modified to provide a very efficient and accurate solution to the problem. A three dimensional error metric is defined and the resulting algorithm is tested for a relatively dense sample for all latitudes, longitudes and altitudes ranging from well under sea level out to ten million kilometers. When using the WGS84 ellipsoidal earth model[18] the maximal (total) error is less than one centimeter over the test region. This level of accuracy may seem excessive given that geodetic earth models represent best fits to the real earth shape and induce far more than one centimeter error. However, when performing simulations the selected earth model is taken as exact. To verify and validate simulations, particularly distributed simulations, it is essential that the coordinate conversions be accurate.

BACKGROUND

A number of reference geodes have been used in astrogeodetic work[18]. These all have the form

(1) $(X / a)^2 + (Y / a)^2 + (Z / c)^2 = 1.$

In this paper the World Geodetic System 1984 (WGS84) is used for the purpose of exposition[18]. For WGS84 a=6,378,137.0 meters and c = 6,356,752.3142 meters. Figure 1 below depicts the geometry of the geocentric (Cartesian) system and the geodetic system in three dimensions.

The geocentric coordinates of a point P are (X,Y,Z) and the corresponding geodetic coordinates of P are (ϕ, λ, h) where ϕ is latitude, λ is longitude and h is the height above the reference ellipsoid. The line connecting the Z axis to P is orthogonal to the tangent plane at the point P_e.

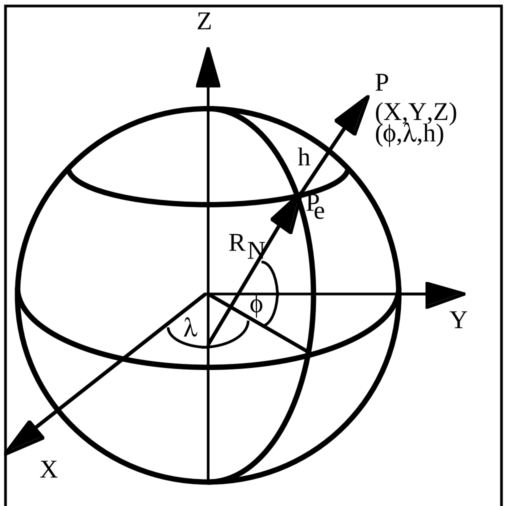

Figure 1

The transformation from geodetic to geocentric coordinates is straight forward[18] and is given by:

(2) $X = (R_N + h) \cos\phi \cos\lambda$

(3) $Y = (R_N + h) \cos\phi \sin\lambda$

(4) $Z = (R_N c^2 / a^2 + h) \sin\phi$

where

(5) $R_N = a / [1 - [\sin^2\phi] (a^2 - c^2) / a^2]^{1/2}.$
The inverse transformation is not as easy and is the subject of this paper. The longitude λ is given by

(6) $\lambda = \tan^{-1}(Y / X)$

and $-\pi/2 \le \lambda \le \pi/2$.

Reference (18) contains other conventions for longitude.

Due to the symmetry of the problem in X and Y it is sufficient to initially work with a meridional section of the geode to determine ϕ and h. This system is depicted in Figure 2.

The meridional ellipse is defined by

(7) $(W / a)^2 + (Z / c)^2 = 1$

and

(8) $W = (X^2+Y^2)^{1/2}$.

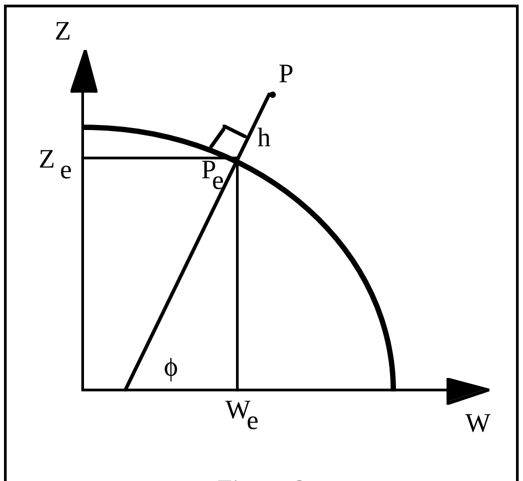

Figure 2

Some useful relations associated with this coordinate system are given below.

The flattening ratio f and the eccentricity e are constants for a particular geode and are defined by

(9) $f = (a - c) / a$

(10) $\varepsilon^2 = (a^2 - c^2) / a^2$.

It is convenient to define an additional constant e´ by

(11) $\varepsilon'^2 = (a^2 - c^2) / c^2$.

Once ϕ has been determined h can be computed from

(12) $h = (W / \cos\phi) - R_N$

for ϕ in non-polar regions. In polar regions it is preferable to use

(13) $h = Z / \sin\phi + R_N(\varepsilon^2 - 1)$,

where R_N is the radius of curvature of the prime vertical and is given by

(14) $R_N = a^2 / (a^2 \cos^2\phi + c^2 \sin^2\phi)^{1/2}$

or equivalently

(15) $R_N = a / (1 - \varepsilon^2 \sin^2\phi)^{1/2}$.

ERROR DEFINITION

Suppose that (X,Y,Z) is the exact location of a point P in the Geocentric Coordinate system. An approximate transformation of the coordinates of P results in another point P_a having approximate geodetic coordinates (ϕ_a, λ_a, h_a). Using the exact relations (2,3,4) the approximate geodetic coordinates can be transformed into corresponding approximate geocentric coordinates (X_a, Y_a, Z_a). The error E induced by the approximation is defined to be the Euclidean distance between P and P_a. That is,

(16) $E = [(X-X_a)^2 + (Y-Y_a)^2 + (Z-Z_a)]^{1/2}$.

E can be viewed as the radius of a ball (sphere) centered at P in the geocentric system.

THE BOWRING ITERATIVE PROCEDURE

In 1976 Bowring[4] developed a very rapidly converging iterative procedure based on Newton's method for computing tan ϕ. The Bowring method is in fact the standard procedure used in the military handbook MIL-HDBK-600008[18] and in Rapp[23,24]. Several of the references[12,29] reject the Bowring method for high speed computation on the basis that the computation of tan ϕ requires several relatively expensive trigonometric function evaluations per step. A simple observation shows in fact that no trigonometric calculations are needed during the iteration. This observation coupled with an improvement in the initial guess yields a very efficient procedure that is so accurate that only one iteration is required. This means that no termination test is needed in most applications.

The Bowring procedure consists of introducing an auxiliary variable ß such that

(17) $\tan \phi_{i+1} = (Z + c\, \varepsilon'^{\,2} \sin^3 ß_i) / (W - a\, \varepsilon^2 \cos^3 ß_i)$

(18) $\tan ß_{i+1} = (1-f) \tan \phi_{i+1}$

with the initial value of ß given by

(19) $\tan ß_0 = aZ / cW.$

The iteration is terminated when $|\tan \phi_{i+1} - \tan \phi_i|$ is small enough and ϕ is then computed by using the inverse tangent function. A cursory examination of (17) and (18) indicates that the sin, cos and inverse tangent must be evaluated for each iteration. As noted in the introduction these evaluations can be avoided and this is the subject of the next section.

THE IMPROVED BOWRING METHOD

Observe that ß does not explicitly appear in equations (17), (18) or (19). Instead sin ß, cos ß and tan ß are required. These terms are readily computed from basic principles. That is, in (17) let

(20) $A_i = Z + c\, \varepsilon'^{\,2} \sin^3 ß_i$

and

(21) $B_i = W - a\, \varepsilon^2 \cos^3 ß_i$

then

(22) $\tan \phi_{i+1} = A_i / B_i.$

Then, by definition,

(23) $\sin \phi_{i+1} = A_i / (A_i^2 + B_i^2)^{1/2}$

(24) $\cos \phi_{i+1} = B_i / (A_i^2 + B_i^2)^{1/2}.$

The values of $\sin ß_{i+1}$ and $\cos ß_{i+1}$ to be used in the next iteration are obtained from equation (18). The initial condition (19) becomes

(25) $\sin ß_0 = aZ / [(aW)^2 + (cZ)^2]^{1/2}$

(26) $\cos ß_0 = cW / [(aW)^2 + (cZ)^2]^{1/2}$

By using (20) to (26) and equation (18) all intervening trigonometric functions are eliminated and replaced with two square roots.

The Bowring method can be further improved by a very simple modification of the initial value of tan ß. Experimentation with the Bowring procedure has shown that the error in tan ϕ is one signed in the first quadrant. Based on this a multiplicative weighting factor is introduced in equation (19) to minimize the error E after one iteration over all points in the first quadrant and for a suitable interval of h values.

NUMERICAL ANALYSIS

Introducing the factor D into (19) yields

(27) $\tan ß_0 = aDZ / cW$

For a specific earth model the value of D can be selected to minimize the error E on a set S that can be determined without knowing ϕ and h. The set S is defined as all points P in the first quadrant with coordinates (W,Z) that lie in the region bounded by the ellipses

(28) $[W / (a+h_{Min})]^2 + [Z / (c+h_{Min})]^2 = 1.$

and

(29) $[W / (a+h_{Max})]^2 + [Z / (c+h_{Max})]^2 = 1.$

The set S is depicted in Figure 3.

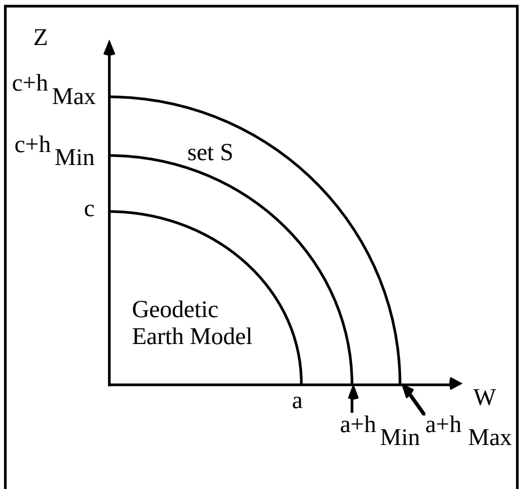

Figure 3

A point (W,Z) is in S if both of the following conditions hold

(30) $[W / (a+h_{Min})]^2 + [Z / (c+h_{Min})]^2 \geq 1$

(31) $[W / (a+h_{Max})]^2 + [Z / (c+h_{Max})]^2 \leq 1.$

For the WGS84 earth model a set of values of h_{Min} , h_{Max} , and D were selected so that the error E is less than 0.01 meters for each region S after one iteration of the improved Bowring procedure. These results are of course dependent on the particular machine environment used and this will be discussed in a later section.

The resulting values of h_{Min}, h_{Max} and aD/c are given in Table 1 below.

Region	h_{Min}	h_{Max}	aD / c
1	$-1.*10^5$	$2.*10^6$	1.0026000
2	$2.*10^6$	$6.*10^6$	1.00092592
3	$6.*10^6$	$18.*10^6$	.999250297
4	$18.*10^6$	$1.*10^9$	.997523508

Table 1: Optimizing Factors

For applications like DIS most of the points will be inside region 1 which extends to 2000 kilometers in altitude. For efficiency the inequalities can be evaluated sequentially in the order given in the table above. In this case only the upper region boundary is evaluated and the inequality can be written in the equivalent form

(32) $$[W]^2 + [Z\,(a+h_{Max}) / (c+h_{Max})]^2 \le (a + h_{Max})^2.$$

This saves a multiply operation.

For a given point P with coordinates X,Y,Z the above equations can be assembled into a step by step procedure for a single iteration. It is assumed that fixed constants such as ε, ε^2,f, and so on are pre-computed.

step 1. Using (8) compute $W= (X^2+Y^2)^{1/2}$.

step 2. Compute Z^2.

step 3. Use equation (32) along with Table 1 to determine which region P is in and thereby determine aD / c.

step 4. Compute $T_0 = Z(aD / c)$.

step 5. Compute $S_0 = [Z(aD / c)]^2 + W^2]^{1/2}$.

step 6. Compute $\sin ß_0 = T_0 /S_0$ and $\cos ß_0 =W /S_0$.

step 7. Compute $T_1 = Z + c\varepsilon'^2 \sin^3 ß_0$.

step 8. Compute $S_1 = \{[T_1]^2 +[W - a\,\varepsilon^2 \cos^3 ß_0]^2\}^{1/2}$.

step 9. Use (23) and (24) to get $\sin \phi_1 = T_1 / S_1$ and $\cos \phi_1 = [W - a\,\varepsilon^2 \cos^3 ß_0] / S_1$.

step 10. Ready to get h. First from (5) get $R_N = a / (1 - \varepsilon^2 \sin^2\phi)^{1/2}$.

step 11. If $|\cos \phi_1| \ge \cos(|67.5 \text{ degrees}|)$ then $h = Z / |\cos \phi_1| - R_N$.

step 12. Else $h = Z / \sin \phi_1 + R_N (\varepsilon-1)$

step 13. Compute ϕ from $\tan^{-1}(\sin \phi_1/\cos \phi_1)$.

step 14. Compute λ from $\tan^{-1}(Y/X)$.

Because only a single iteration is used equation (18) is not needed. If a second iteration is desired step 9 needs to be modified to account for (18).

This algorithm will fail if $\phi = \pi/2$ or $-\pi/2$ (or is very close to one of these values). In this case X is zero (or nearly zero) and both ϕ and h are known immediately. In implementing the algorithm in software it is important to include tests for these cases and take the appropriate action. With such a test in place the single-argument inverse tangent routine can be used in step 13. In step 14. similar tests must be implemented depending on the convention for latitude.

Note that only the squares of X and Y are involved in the procedure and W is positive. The sign of Z then determines the sign of ϕ, so that the procedure yields the proper value of ϕ for all quadrants. Note that the test on 67.5 degrees latitude defines the region between the Arctic and Antarctic circles.

All of the square root evaluations and the inverse tangent evaluations can be eliminated from the above procedure by using in-line code. Whether this should be done or not depends on the particular computer environment being used. Older (legacy) environments usually have their mathematical functions implemented in software. In such cases, repeated subroutine calls can be relatively expensive[27] and there is a substantial payoff in terms of reduced execution time when in-line routines are implemented. It should be noted that some legacy machines, such as the CRAY family, have a hardware implementation of the inverse square root that is used as a basis for very efficient transcendental subroutines. For most current workstations the mathematical routines are embedded in a Floating Point Unit (FPU) to provide efficient processing. Most square root implementations on FPU equipped machines take the equivalent of about five floating point operations per call. However, there is a substantial variation in the processing time for the transcendental functions from one machine to another. Generally, when using an FPU equipped system, in-line code for the mathematical functions is not competitive with the built in mathematical routines. One exception to this is given below where the square root in step 10. is evaluated using an in-line procedure.

In-line square root evaluation involves the use of Newton's method for finding square roots and has been used to great advantage in embedded systems[28]. The following sequence converges to the square root of A given an accurate enough initial value[9]

(33) $x_{i+1} = .5(x_i + A/x_i)$.

The term $(1 - \varepsilon^2 \sin^2\phi)^{1/2}$ of step 10. can be expanded in a rapidly convergent binomial series because the term $U = \varepsilon^2 \sin^2\phi$ is very small. A two term binomial series for the initial guess coupled with one iteration of (33) along with some simplification yields

(34) $V = 0.5 - 0.25U$

(35) $(1 - \varepsilon^2 \sin^2\phi)^{1/2} \sim V + (V - 0.25)/V$.

Use of equations (34) and (35) is at least as fast as calling a square root routine even when an FPU is being used and has no impact on the overall accuracy of the basic algorithm.

For machines that do not have a fast square root function the square root in step 8. can also be eliminated. This involves the use of a sequential square root method based on two iterations of (33)[28]. The value of S_0 computed in step 5. is used as an initial guess for two iterations of (33). This approach has no effect on the total error for h less than a million kilometers.

ERROR EVALUATION

The number representation of the machine on which the experimental calculations were performed has a 23 bit mantissa. Single precision on such a machine will lose about a meter of accuracy in representing numbers like a and c. As a consequence double precision was used for all calculations. For machines with extended word lengths double precision will not be an issue.

To assess the error in the algorithm a test program was developed that defined very dense sets of exact points (h,ϕ,) on a rectangular grid. These points were converted exactly by (2), (3) and (4) into a corresponding set of geocentric points (X,Y,Z). The fourteen step algorithm was applied to obtain a set of approximate points and the error E was determined. The maximum error over the entire region was recorded. In no case did the error exceed one centimeter on the region encompassing all latitudes and longitudes for all h (in meters) in $[-10^5, 10^{11}]$. When step 14. was modified to use (34) and (35) the error was still less than one centimeter over the same region. When the square root in (8) was replaced as suggested above the error was less than one centimeter for h in $[-10^5, 10^9]$.

TIMING ESTIMATES

The only way to really assess run time is to implement the algorithm on a particular machine and test it. However, some idea of the relative cost of the algorithm can be obtained by using operation counts. This permits comparisons between algorithms that are less machine dependent.

The paper by Wise[29] uses such an approach and provides a convenient means of comparison. Wise used the number of floating point operations, floats for short, as a measure of computational cost. He assumed that multiply, divide and add all take the same computational time. This is a relatively good assumption but is clearly machine dependent. Based on some empirical evidence he concluded that it took five floats for a square root and twelve for the trigonometric functions. The algorithm proposed by Wise was then estimated to take 56 + 44i floats where i is the number of iterations used. He assumed that reasonable care was taken in the programming process to eliminate redundant calculations. In a later paper[12] Lin and Ng proposed a similar procedure that had an estimated computational cost that was 20 percent less than the Wise algorithm. This would result in 0.8(56 + 44i). The error criterion used in both papers was to achieve a 50 centimeter accuracy in h and there was little or no discussion of the effect of the angular errors. It should be noted that the error criterion used in this paper guarantees that the error in h is less than one centimeter.

The computational cost of the fourteen step procedure of this paper was estimated using the assumptions made by Wise. Logical tests were ignored because there are very few of them and because they are relatively fast compared to floats. It is presumed that Wise made a similar assumption. Because of the logical tests the run time will depend on the location of P. To simplify the analysis it is assumed that the altitude of P is less than 2000 kilometers and that P lies between the Arctic and Antarctic circles. The resulting total number of floats is 78. Table 2 below shows the comparison with the algorithms mentioned above.

i	56 +44i	0.8(56 +44i)
1	100	80
2	144	115
3	188	150

Table 2: Computational Cost Estimates

Wise presented examples for his procedure that meet the 50 centimeter error (in h) requirement. Between -15,000 and 17,000 meters one iteration was sufficient. Between 17,000 meters and 350,000 meters two iterations were required and commented that this operating range would suffice for all aircraft and many ballistic missiles. Between 350,000 and a million meters it took three iterations and this suffices for medium altitude orbits. He also noted that for some locations it took an extra iteration. Similar behavior could be expected for the approach of Lin and Ng.

CONCLUSIONS

The algorithm developed in this paper yields far more accurate results than most of the published algorithms for what is apparently less computational cost. The algorithm is also robust, and is for practical purposes, applicable for all latitudes and altitudes. A significant attribute of the procedure is that the processing time is nearly a constant.

The computational cost of the algorithm probably can be further reduced with a corresponding loss in accuracy. For example, the binomial series can be used in step 10. without the iterative correction. Further efficiencies have not been pursued in this paper because the permissible error depends on the application.

It should be noted that other published iterative procedures[3,26] that appear to involve trigonometric functions could be treated as in this paper. Some experiments were conducted using other geodetic earth models without changing the optimizing coefficients of Table 1. The errors that resulted were also on the order of one centimeter.

REFERENCES

1. Bartelme, N. and Messily, P., *Ein einfaches, rasches und numerisch stabiles Verfahren zur Bestimmung des kurzesten Abstandes eines Punktes von einen spharoidischen Rotations-ellipsoid,* AVN, Heft 12, pp 436-439 (Wichmann, Karisruhe).

2. Borkowski, K. M., "Transformation of Geocentric to Geodetic Coordinates Without Approximations," *Astrophys. Space Science,* 139, pp 1-4, 1987 and 146, pp 201, 1988.

3. Borkowski, K. M., "Accurate Algorithms to Transform Geocentric to Geodetic Coordinates," *Bulletin Geodesique,* 63, 1989, pp. 50-56.

4. Bowring, B.R., "Transformation from Spatial to Geophysical Coordinates," *Survey Rev.*, V 23, 181, 323-327, 1976.

5. Brooks, Rita M., *"Coordinate Transformation Formulas,"* Technical Note No. 3280-220, Test Data Division and Land-Air, Inc., Pacific Missile Range, Nov. 1962.

6. Burchfiel, J. and Smyth, S., "Use of Global Coordinates in the SIMNET Protocol," White Paper ASD-90-10, Second Workshop on Standards for Interoperability of Defense Simulations, Orlando, Fl., January 1990.

7. Chauvenet, W., *A Manual of Spherical and Practical Astronomy, Vol. I- Spherical Astronomy.* Fifth Ed., Dover Publishing Inc., 1960.

8. "Draft, Communication Architecture for Distributed Interactive Simulation," Technical Report IST-CR-92-6. Institute for Simulation and Training, University of Central Florida, Orlando Florida, May 1992.

9. Hart, J. F., Cheney, E. W., Lawson, C. L., Maehly, H. J., Meesztenyi, C. K., Rice, J. R., Thacher, H. G., and Witzgall, C., *Computer Approximations,* John Wiley & Sons, N. Y.,1978.

10. Hedgley, D. R., *An Exact Transformation from Geocentric to Geodetic Coordinates for Nonzero Altitudes,* NASA Technical Report, R-458, 1976.

11. Heiskanen, W. A and Vening,Meinez, F. A., *The Earth and its Gravitational Field,* McGraw-Hill, 1958.

12. Kuo-Chi Lin and Huat Keng Ng, *Interconversions Between Different Coordinate Systems,* White Paper, Institute for Simulation and Training, University of Central Florida, Orlando Florida, 1992.

13. Kuo-Chi Lin and Huat Ng, "Coordinate Transformations in Distributed Interactive Simulation (DIS)", *SIMULATION,* Vol. 61, No. 5, 1993, pp. 326-331.

14. *Lagrangian Dynamics*, Schaum Publishing Co., NY, pp. 281-285.

15. Lambent, W. D. and Sick, C. H., F*ormulas and Tables for the Computation of Geodetic Positions on the International Ellipsoid*, SPELL. Pub. No. 200, U. S. Coast and Geodetic Survey, 1935.

16. Long, S., A., T., *Derivation of Transformation Formulas Between Geocentric and Geodetic Coordinates for Nonzero Altitudes*, NASA Technical Note, Langley Research Center, NASA TN D-7522, July 1974.

17. Long, S., A., T., "General Altitude Transformation Between Geocentric and Geodetic Coordinates," *Celestial Mechanics*, 12, pp. 225-230, 1975.

18. Military Handbook, *Datums, Projections, Grids and Common Coordinate Systems, Transformation Of,* US Department of Defense, MIL-HDBK-600008.

19. *Military Standard (Final Draft) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation,* Technical Report IST-PD-91-1., Institute for Simulation and Training, University of Central Florida, Orlando Florida, May 1992.

20. Morrison, J. and Pines, S., "The Reduction from Geocentric to Geodetic Coordinates," *Astronomical Journal,* 66, pp. 15-16, 1961.

21. Paul, M. K., "A Note on Computation of Geodetic Coordinates from Geocentric (Cartesian) Coordinates," *Bulletin, Geod.. Nouv. Ser.,* No. 108, pp. 135-139, 1973.

22. Pick, M., "Closed Formulae for Transformation of the Cartesian Coordinate System into a System of Geodetic Coordinates," *Studia Geoph. et Geod.,* 29, pp. 112-119, 1985.

23. Rapp, Richard H., *Geometric Geodesy - Part I,* Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio, 1984.

24. Rapp, Richard H., *Geometric Geodesy - Part II*, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio, 1987

25. Sofair, I., *An Improved Method for Calculating the Exact Geodetic Latitude and Altitude*, NSWCDD/TR-94/77, Naval Surface Warfare Center, Dahlgren, VA., 1994.

26. *The Astronomical Almanac for the Year 1995*, U. S. Government Printing Office, Wash. D. C., 1995, Appendix K.

27. Toms, R. M., *An Empirical Study of Normalized Timing Requirements for Various Standard Operations*, Lawrence Livermore National Laboratory, University of California, UCID - 21107, June 1987.

28. Toms, R. M., *A Bomb Trajectory Algorithm for Strategic Aircraft,* Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, pp. 610-615, (1976).

29. Wise, B., *Geocentric to Geodetic Coordinate Conversions,* BBN Report No. 7756, May 1992.

This work was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.